

# METHOD AND SYSTEM FOR INCORPORATING SECURITY MECHANISM INTO SESSION INITIATION PROTOCOL REQUEST MESSAGE FOR CLIENT PROXY AUTHENTICATION

**Patent number:** JP2003108527 (A)  
**Publication date:** 2003-04-11  
**Inventor(s):** BOBDE NIKHIL P; DEMIRTJIS ANN; HAN MU  
**Applicant(s):** MICROSOFT CORP  
**Classification:**  
- international: G06F21/20; G06F15/00; G09C1/00; H04L29/06; G06F21/20; G06F15/00; G09C1/00; H04L29/06; (IPC1-7): G06F15/00; G09C1/00  
- european: H04L29/06S12A; H04L29/06C2; H04L29/06M2H2; H04L29/06S8A  
**Application number:** JP20020174951 20020614  
**Priority number(s):** US20010298239P 20010614; US20020151747 20020517

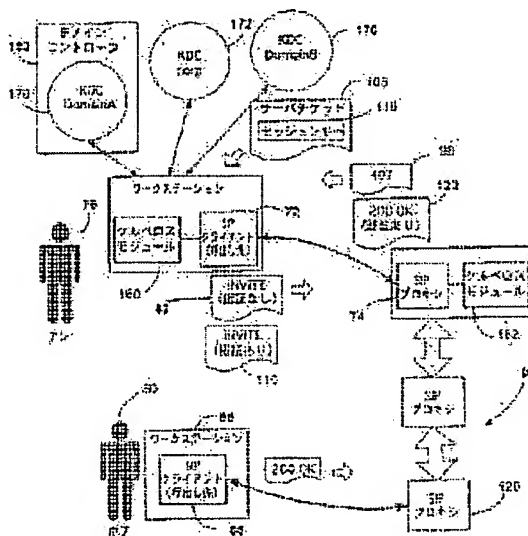
## Also published as:

JP4294268 (B2)  
EP1267548 (A2)  
EP1267548 (A3)  
EP1267548 (B1)  
US2008022383 (A1)

more >>

## Abstract of JP 2003108527 (A)

**PROBLEM TO BE SOLVED:** To provide a method and a system for allowing an SIP client and an SIP proxy to authenticate each other by incorporating a Cerberus security mechanism into a message flow of signaling operation based on a session initiation protocol. **SOLUTION:** When receiving a request message such as an INVITE request from the SIP client, the SIP proxy sends a challenge message for indicating the necessity of authentication based on Cerberus in response to this message. The SIP client sends a second request message having a proxy authorization header including authenticating data in response to this message so that the proxy can authenticate a user of the client.



Data supplied from the esp@cenet database — Worldwide

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号  
特開2003-108527  
(P2003-108527A)

(43) 公開日 平成15年4月11日 (2003.4.11)

(51) Int.Cl. <sup>7</sup>	識別記号	F I	テームコード* (参考)
G 0 6 F 15/00	3 3 0	G 0 6 F 15/00	3 3 0 B 5 B 0 8 5
	3 1 0		3 1 0 D 5 J 1 0 4
G 0 9 C 1/00	6 4 0	G 0 9 C 1/00	6 4 0 E

審査請求 未請求 請求項の数28 O L (全 23 頁)

(21) 出願番号 特願2002-174951(P2002-174951)

(22) 出願日 平成14年6月14日 (2002.6.14)

(31) 優先権主張番号 60/298,239

(32) 優先日 平成13年6月14日 (2001.6.14)

(33) 優先権主張国 米国 (U S)

(31) 優先権主張番号 10/151,747

(32) 優先日 平成14年5月17日 (2002.5.17)

(33) 優先権主張国 米国 (U S)

(71) 出願人 391055933

マイクロソフト コーポレーション  
MICROSOFT CORPORATI  
ON  
アメリカ合衆国 ワシントン州 98052-  
6399 レッドモンド ワン マイクロソフ  
ト ウェイ (番地なし)

(72) 発明者 ニキル ビー. ポプデ

アメリカ合衆国 98052 ワシントン州  
ベルビュー 43 プレイス 14405

(74) 代理人 100077481

弁理士 谷 義一 (外2名)

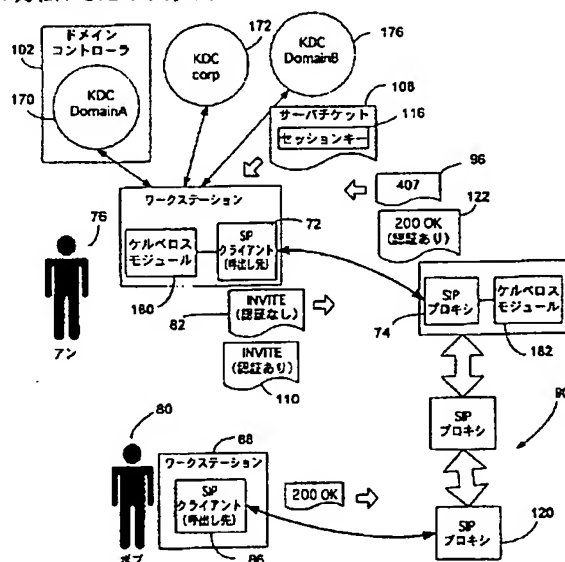
最終頁に続く

(54) 【発明の名称】 クライアント・プロキシ認証のためにセッションイニシエーションプロトコルリクエストメッセージにセキュリティ機構を組み込むための方法およびシステム

(57) 【要約】

【課題】 セッションイニシエーションプロトコルに基づくシグナリングオペレーションのメッセージフローにケルベロスセキュリティ機構を組み込んで、SIPクライアントとSIPプロキシが相互に認証することを可能にする方法およびシステムを提供する。

【解決手段】 INVITEリクエストなどのリクエストメッセージをSIPクライアントから受け取ると、SIPプロキシはこれに回答して、ケルベロスに基づく認証が必要であることを指示するチャレンジメッセージを送る。これに回答してSIPクライアントは、プロキシに対するケルベロスサーバチケットを含む認証データを含んだproxy authorizationヘッダを有する第2のリクエストメッセージを送って、プロキシがクライアントのユーザを認証できるようにする。



## 【特許請求の範囲】

【請求項1】 セッションイニシエーションプロトコル(SIP)クライアントのユーザを認証するために、SIPプロキシによりステップを実行するためのコンピュータ実行可能命令を有するコンピュータ可読媒体であって、前記ステップが、

前記SIPクライアントから第1のリクエストメッセージを受け取るステップと、

前記第1のリクエストメッセージが、前記SIPクライアントのユーザを認証するための認証データを含んでいないと判定するステップと、

認証が必要であることを指示するコードを含むチャレンジメッセージを送るステップと、

選択された認証プロトコルに従って前記SIPクライアントのユーザを認証するための認証データを含むproxy-authorizationヘッダを含む第2のリクエストメッセージを前記SIPクライアントから受け取るステップと、

前記第2のリクエストメッセージのproxy-authorizationヘッダの中の認証データを使用して前記SIPクライアントのユーザを認証するステップとを備えることを特徴とするコンピュータ可読媒体。

【請求項2】 請求項1に記載のコンピュータ可読媒体であって、前記第1および第2のリクエストメッセージがSIP INVITEリクエストであることを特徴とするコンピュータ可読媒体。

【請求項3】 請求項1に記載のコンピュータ可読媒体であって、前記SIPクライアントのユーザの認証に成功した後に、前記第2のリクエストメッセージを、前記リクエストメッセージの中で識別される意図された呼出し先に至るSIPシグナリング経路に転送するステップを実行するためのコンピュータ実行可能命令をさらに有することを特徴とするコンピュータ可読媒体。

【請求項4】 請求項1に記載のコンピュータ可読媒体であって、前記選択された認証プロトコルがケルベロスプロトコルであり、前記proxy-authorizationヘッダの中の認証データが、前記SIPプロキシにアクセスするためのケルベロスサーバチケットを表すデータを含むことを特徴とするコンピュータ可読媒体。

【請求項5】 請求項4に記載のコンピュータ可読媒体であって、前記認証ステップが、ケルベロスモジュールを呼び出して前記ケルベロスサーバチケットの有効性をチェックするステップと、前記ケルベロスサーバチケットから、前記SIPクライアントと通信するのに使用するセッションキーを抜き出すステップとを含むことを特徴とするコンピュータ可読媒体。

【請求項6】 請求項1に記載のコンピュータ可読媒体であって、前記proxy-authorizationヘッダの中の認証データが、前記SIPクライアント

と前記SIPプロキシとの間の相互認証を要求するデータを含み、前記コンピュータ可読媒体が、前記SIPクライアントが前記SIPプロキシを認証するのに使用する前記SIPプロキシの認証データを含むproxy-authentication informationヘッダを有するメッセージを、前記SIPクライアントに返すステップを実行するコンピュータ実行可能命令をさらに有することを特徴とするコンピュータ可読媒体。

【請求項7】 請求項1に記載のコンピュータ可読媒体であって、前記選択された認証プロトコルがNTLMプロトコルであることを特徴とするコンピュータ可読媒体。

【請求項8】 セッションイニシエーションプロトコル(SIP)クライアントが、SIPプロキシを介したセッションの開始に関連して前記SIPクライアントのユーザを前記SIPプロキシに対して認証するステップを実行するためのコンピュータ実行可能命令を有するコンピュータ可読媒体であって、前記ステップが、意図された呼出し先に対する第1のリクエストメッセージを前記SIPプロキシに送るステップと、

前記第1のリクエストメッセージに対する応答として前記SIPプロキシから送られた、認証が必要であることを指示するチャレンジメッセージを受け取るステップと、

選択された認証プロトコルに従って前記ユーザを認証するための認証データを含むproxy-authorizationヘッダを構築するステップと、前記構築されたproxy-authorizationヘッダを含む、前記意図された呼出し先に対する第2のリクエストメッセージを送るステップとを備えることを特徴とするコンピュータ可読媒体。

【請求項9】 請求項8に記載のコンピュータ可読媒体であって、前記第1および第2のリクエストメッセージがSIP INVITEリクエストであることを特徴とするコンピュータ可読媒体。

【請求項10】 請求項8に記載のコンピュータ可読媒体であって、前記選択された認証プロトコルがケルベロスプロトコルであり、前記proxy-authorizationヘッダの中の認証データが、前記SIPプロキシにアクセスするためのケルベロスサーバチケットを表すデータを含むことを特徴とするコンピュータ可読媒体。

【請求項11】 請求項8に記載のコンピュータ可読媒体であって、前記proxy-authorizationヘッダを構築するステップが、ケルベロスキーディストリビューションセンターから前記ケルベロスサーバチケットを取得するステップを含むことを特徴とするコンピュータ可読媒体。

【請求項12】 請求項11に記載のコンピュータ可読

媒体であって、前記proxy-authenticationヘッダが、前記SIPクライアントと前記SIPプロキシとの間の相互認証のリクエストを表すデータを含み、前記コンピュータ可読媒体が、前記第2のリクエストメッセージに対する応答として前記SIPプロキシから応答メッセージを受け取るステップと、

前記応答メッセージの中に含まれるproxy-authentication informationヘッダから、前記SIPプロキシの認証データを抜き出すステップと、

前記proxy-authentication informationヘッダから抜き出した前記SIPプロキシの認証データに基づいて前記SIPプロキシを認証するステップとを実行するためのコンピュータ実行可能命令をさらに含むことを特徴とするコンピュータ可読媒体。

【請求項13】 請求項8に記載のコンピュータ可読媒体であって、前記SIPクライアントが、前記選択された認証プロトコルに従って前記SIPクライアントのユーザを認証するためのユーザ認証データを取得するステップと、

前記SIPプロキシに登録するために、前記ユーザを認証するための認証データを含むproxy-authenticationヘッダを有するREGISTERメッセージを前記SIPプロキシに送信するステップとを実行するためのコンピュータ実行可能命令をさらに有することを特徴とするコンピュータ可読媒体。

【請求項14】 請求項13に記載のコンピュータ可読媒体であって、前記選択された認証プロトコルがケルベロスプロトコルであり、前記ユーザの認証データが、ケルベロスキーディストリビューションセンターから取得した、前記SIPプロキシにアクセスするためのケルベロスサーバチケットを表すデータを含むことを特徴とするコンピュータ可読媒体。

【請求項15】 請求項8に記載のコンピュータ可読媒体であって、前記選択された認証プロトコルがNTLMプロトコルであることを特徴とするコンピュータ可読媒体。

【請求項16】 セッションイニシエーションプロトコル(SIP)プロキシが、セッション開始オペレーションの間にSIPクライアントのユーザを認証するための方法であって、

前記SIPクライアントから第1のリクエストメッセージを受け取るステップと、

前記第1のリクエストメッセージが、前記SIPクライアントのユーザを認証するための認証データを含んでいないと判定するステップと、

「407 Proxy Authentication Required」ステータスコードを含むメッセ

ジを前記SIPクライアントに送って、認証が必要であることを指示するステップと、

前記SIPクライアントのユーザを認証するためのユーザ認証データを含むproxy-authenticationヘッダを含む第2のリクエストメッセージを前記SIPクライアントから受け取るステップであって、前記ユーザ認証データが、前記SIPプロキシにアクセスするためのケルベロスサーバチケットを表すデータを含むステップと、

前記ケルベロスサーバチケットを使用して前記SIPクライアントのユーザを認証し、前記SIPクライアントとの通信を暗号化するためのセッションキーを前記ケルベロスサーバチケットから抜き出すステップと、

前記第2のリクエストメッセージを、前記INVITEメッセージの中で識別される意図された呼出し先に至るSIPシグナリング経路に転送するステップとを備えることを特徴とする方法。

【請求項17】 請求項16に記載の方法であって、前記第1および第2のリクエストメッセージがSIP INVITEリクエストであることを特徴とする方法。

【請求項18】 請求項16に記載の方法であって、前記第2のリクエストメッセージの中のproxy-authenticationヘッダの認証データが、前記SIPクライアントと前記SIPプロキシとの間の相互認証を要求するデータを含み、前記方法が、前記SIPクライアントが前記SIPプロキシを認証するのに使用する認証データを含むproxy-authentication informationヘッダを有するメッセージを前記SIPクライアントに返すステップをさらに備えることを特徴とする方法。

【請求項19】 セッションイニシエーションプロトコル(SIP)クライアントが、SIPプロキシを介したセッションの開始に関連して前記SIPクライアントのユーザを前記SIPプロキシに対して認証するための方法であって、

意図された呼出し先に対する第1のリクエストメッセージを前記SIPプロキシに送るステップと、

前記第1のリクエストメッセージに対する応答として前記SIPプロキシにより送られた、認証が必要であることを指示するチャレンジメッセージを受け取るステップと、

前記ユーザを認証するためのユーザ認証データを含むproxy-authenticationヘッダを構築するステップであって、前記ユーザ認証データが、前記SIPプロキシにアクセスするためのケルベロスサーバチケットを表すデータを含むステップと、

前記構築されたproxy-authenticationヘッダを含む、前記意図された呼出し先に対する第2のリクエストメッセージを送るステップとを備えることを特徴とする方法。

【請求項20】 請求項19に記載の方法であって、前記proxy-authORIZATIONヘッダを構築するステップが、ケルベロスキーディストリビューションセンターから前記ケルベロスサーバチケットを取得するステップを含むことを特徴とする方法。

【請求項21】 請求項19に記載の方法であって、前記proxy-authORIZATIONヘッダを構築するステップが、前記SIPクライアントと前記SIPプロキシとの間の相互認証のリクエストを、前記proxy-authORIZATIONヘッダに挿入するステップを含み、前記方法が、  
前記第2のリクエストメッセージに対する応答として前記SIPプロキシから応答メッセージを受け取るステップと、  
前記応答メッセージの中に含まれるproxy-authentication informationヘッダから、前記SIPプロキシの認証データを抜き出すステップと、  
前記proxy-authentication informationヘッダから抜き出した前記SIPプロキシの認証データに基づいて前記SIPプロキシを認証するステップとをさらに備えることを特徴とする方法。

【請求項22】 請求項21に記載の方法であって、前記第1および第2のリクエストメッセージがSIP INVITEリクエストであることを特徴とする方法。

【請求項23】 セッションイニシエーションプロトコル(SIP)クライアントが、SIPプロキシと認証を実行するための方法であって、  
ケルベロス認証プロトコルに従って前記SIPクライアントを認証するための認証データを取得するステップであって、前記認証データが、前記SIPプロキシにアクセスするためのサーバチケットを含むステップと、  
前記SIPプロキシに登録するために、前記認証データを含むproxy-authORIZATIONヘッダを有するREGISTERメッセージを前記SIPプロキシに送信するステップとを備えることを特徴とする方法。

【請求項24】 請求項23に記載の方法であって、前記proxy-authORIZATIONヘッダが、  
前記SIPサーバとの相互認証のリクエストを含み、前記方法が、  
前記REGISTERメッセージに対する応答として前記SIPプロキシから応答メッセージを受け取るステップと、  
前記応答メッセージの中に含まれるproxy-authentication informationヘッダから、前記SIPプロキシの認証データを抜き出すステップと、  
前記proxy-authentication in

formationヘッダから抜き出した前記SIPプロキシの認証データに基づいて前記SIPプロキシを認証するステップとをさらに備えることを特徴とする方法。

【請求項25】 セッションイニシエーションプロトコル(SIP)リクエストメッセージを表すデータ構造が記憶されたコンピュータ可読媒体であって、  
SIPプロキシにアクセスするためのケルベロスサーバチケットを表すデータを含むデータフィールドを有するproxy-authORIZATIONヘッダを含む複数のSIPヘッダと、  
メッセージ本体とを備えることを特徴とするコンピュータ可読媒体。

【請求項26】 請求項25に記載のコンピュータ可読媒体であって、前記proxy-authORIZATIONヘッダが、前記ケルベロスサーバチケットに関連するセッションキーを使用して前記SIPリクエストメッセージの一部に署名することによって生成された署名を有する第2のデータフィールドを有することを特徴とするコンピュータ可読媒体。

【請求項27】 請求項25に記載のコンピュータ可読媒体であって、前記SIPリクエストメッセージがSIP INVITEリクエストであることを特徴とするコンピュータ可読媒体。

【請求項28】 請求項25に記載のコンピュータ可読媒体であって、前記SIPリクエストメッセージがSIP REGISTERリクエストであることを特徴とするコンピュータ可読媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は一般に、コンピュータネットワークを介した装置間通信に関し、詳細には、セキュリティ機構、例えばケルベロス認証プロトコルに基づくセキュリティ機構を、セッションイニシエーションプロトコル(SIP: Session Initiation Protocol)を通信セッションを確立するためのシグナリングプロトコルとして使用したネットワーク通信に組み込むことに関する。

【0002】

【従来の技術】セッションイニシエーションプロトコル(SIP)は、コンピューティング装置が、自体がコンピュータネットワークを介して通信したいと欲している他の装置の位置を突き止め、その装置との間に通信セッションを確立するための機構を提供するシグナリングプロトコルである。SIPは汎用性のあるプロトコルであり、多くの異なるシナリオで通信セッションを確立するのに使用されてきた。SIPは例えば、インターネット会議、電話、プレゼンス(presence)、イベント通知およびインスタントメッセージングに使用されている。SIPの重要な強みは、位置独立の単一のアドレ

スの下にいる被呼者（ユーザ）に、たとえこの被呼者が別のコンピュータに移動していたとしてもたどり着くことができる、パーソナルモビリティ（personal mobility）をサポートしていることである。

【0003】SIPに基づくセッション開始オペレーションの共通する1つのモードは「プロキシ（proxy）モード」である。例えば、SIPクライアント（「呼出し元」）は、意図された受信者（「呼出し先」）を電子メールアドレスに似たアドレスによって識別するINVITEメッセージなどのSIPリクエストメッセージを送ることができる。このリクエストメッセージは一般にまず、送信側SIPクライアントのアウトバウンド（outbound）SIPプロキシに送られる。次いでこのアウトバウンドSIPプロキシがこのリクエストメッセージを、しばしば他の中間SIPプロキシを介して、意図された受信側クライアントが登録されているSIPプロキシに転送し、次いでこのSIPプロキシが、INVITEメッセージを受信側クライアントに送る。受信側クライアントの受入れメッセージ（「200OK」）が、シグナリングチェーンを介して呼出し元クライアントに戻され、これによって、呼出し元クライアントは呼出し先クライアントと、一般にシグナリングチャンネルとは別の媒体チャンネルを通して通信できるようになる。他のSIPクライアントと通信する他に、SIPクライアントはさらに、REGISTERリクエストを送ることによってSIPレジストラ（registrar）に自体を登録するなどの目的で、SIPサーバと通信することができる。

【0004】

【発明が解決しようとする課題】さまざまな応用に対して幅広く実装されているが、SIPは、主としてシグナリングオペレーション用に設計されたものである。SIPは、通信セッションのセキュリティおよびプライバシーを保護するためのセキュリティ機構を明示的に提供せず、またはこれを必要としない。しかし多くのケースで、SIPクライアントが、SIPクライアントのユーザを認証するよう求めるリクエストをアウトバウンドSIPプロキシに送ることを要求し、アウトバウンドSIPプロキシが、SIPクライアントに対して自体を認証することを要求することが望ましい。さらに、SIPリクエストメッセージの完全性を保護することもしばしば必要である。クライアント・プロキシ認証およびメッセージの完全性はともに、信頼性の高いセキュリティ機構の使用を要求する。したがって、信頼性の高いセキュリティ機構をSIPシグナリングオペレーションと組み合わせ、SIPクライアントとアウトバウンドプロキシとの間の認証を可能にすることが求められている。しかし、所望のセキュリティ機構をSIPシグナリングフレームワークにどのようにはめ込んで、目的の異なるこれらの2つの機構を一緒に効果的に実行することができる

ようにするかということが技術上の課題である。

【0005】

【課題を解決するための手段】以上のことを考慮して、本発明は、ケルベロス（Kerberos）プロトコル、NTLMプロトコルなどのセキュリティ機構を、SIPシグナリングオペレーションのメッセージフローに組み込んで、SIPクライアントとSIPプロキシが相互に認証できるようにするための方式を提供する。本発明によれば、SIPクライアントからSIPリクエストメッセージを受け取ると、プロキシはこれにตอบสนองして、予め選択されたセキュリティ機構に基づく認証が必要であることを指示するチャレンジメッセージを送る。これにตอบสนองしてSIPクライアントは、前記セキュリティ機構に基づいてサーバに対してクライアントを認証するための認証データを含むproxy authorizationヘッダを有する前記リクエストメッセージの第2バージョンないし改訂バージョンを送る。ケルベロスセキュリティ機構を使用する場合には、proxy authorizationヘッダが、プロキシにアクセスするためにクライアントが取得したケルベロスサーバチケットを表すデータを含む。proxy authorizationヘッダのデータに基づくクライアントのユーザの認証に成功した場合、SIPプロキシは、SIPクライアントとリクエストメッセージの意図された受信者との間のSIPメッセージシグナリング経路に沿ってこのリクエストを転送する。SIPクライアントが相互認証を要求している場合、SIPプロキシは、クライアントに送る次のメッセージにproxy authentication information（プロキシ認証情報）ヘッダを追加する。このメッセージは例えば、INVITEリクエストにตอบสนองして呼出し先SIPクライアントが生成した「200OK」SIP応答、またはREGISTERメッセージにตอบสนองしてSIPレジストラサーバが生成した「200OK」応答である。proxy authentication informationヘッダは、クライアントがSIPプロキシを認証するための認証データを含む。

【0006】本発明の特徴は請求項に詳細に記述されているが、本発明、ならびに本発明の目的および利点は、添付図面を参照して以下の詳細な説明を読むことによって最もよく理解されよう。

【0007】

【発明の実施の形態】図面を参照すると、適当なコンピューティング環境中に実現された本発明が示されている。図面中、同様の参照符号は同様の要素を指す。そうでなければならないというわけではないが、本発明は、パーソナルコンピュータによって実行される、プログラムモジュールなどのコンピュータ実行可能命令の一般的な文脈で説明される。プログラムモジュールには一般に、特定のタスクを実行し、または特定の抽象データ型

を実装するルーチン、プログラム、オブジェクト、コンポーネント、データ構造などが含まれる。さらに、本発明を、ハンドヘルド装置、マルチプロセッサシステム、マイクロプロセッサベースの、またはプログラム可能な家庭用電子機器、ネットワークPC、ミニコンピュータ、メインフレームコンピュータなどを含む他のコンピュータシステム構成とともに実施できることを当業者は理解されたい。本発明は、通信ネットワークを介してリンクされた遠隔処理装置によってタスクが実行される分散コンピューティング環境で実施することができる。分散コンピューティング環境では、プログラムモジュールが、ローカルメモリ記憶装置と遠隔メモリ記憶装置の両方に位置することができる。

【0008】以下の説明ではまず、本発明を実現するための例示的なシステム中で使用することができる汎用コンピューティング装置を説明し、次いで、図2〜9を参照して本発明をより詳細に説明する。図1を参照すると、処理ユニット21、システムメモリ22およびシステムバス23を含む従来のパーソナルコンピュータ20の形態の汎用コンピューティング装置が示されている。システムバス23は、システムメモリを含むさまざまなシステムコンポーネントを処理ユニット21に結合する。システムバス23は、さまざまなバスアーキテクチャのうちの任意のアーキテクチャを使用した、メモリバスまたはメモリコントローラ、周辺バスおよびローカルバスを含むいくつかのタイプのバス構造のうちの任意の構造とすることができる。システムメモリは、リードオンリーメモリ（ROM）24およびランダムアクセスメモリ（RAM）25を含む。ROM24には、起動時などにパーソナルコンピュータ20の内部要素間の情報転送を助ける基本ルーチンを含む基本入出力システム（BIOS）26が記憶されている。パーソナルコンピュータ20はさらに、ハードディスク60の読取り/書込み用のハードディスクドライブ27、リムーバブル磁気ディスク29の読取り/書込み用の磁気ディスクドライブ28、およびCD-ROMまたは他の光媒体などのリムーバブル光ディスク31の読取り/書込み用の光ディスクドライブ30を含む。

【0009】ハードディスクドライブ27、磁気ディスクドライブ28および光ディスクドライブ30はそれぞれ、ハードディスクドライブインタフェース32、磁気ディスクドライブインタフェース33、および光ディスクドライブインタフェース34によってシステムバス23に接続されている。これらのドライブおよび関連コンピュータ可読媒体は、コンピュータ可読命令、データ構造、プログラムモジュール、およびパーソナルコンピュータ20のためのその他のデータを記憶する不揮発性記憶装置を提供する。ここに記載した例示的な環境では、ハードディスク60、リムーバブル磁気ディスク29およびリムーバブル光ディスク31が使用されているが、

この例示的な動作環境では、磁気カセット、フラッシュメモ리카ード、デジタルビデオディスク、ベルヌーイカートリッジ、ランダムアクセスメモリ、リードオンリーメモリ、ストレージエリアネットワークなど、コンピュータがアクセス可能なデータを記憶することができる他のタイプのコンピュータ可読媒体を使用することもできることを当業者は理解されたい。

【0010】ハードディスク60、磁気ディスク29、光ディスク31、ROM24またはRAM25には、オペレーティングシステム35、1つまたは複数のアプリケーションプログラム36、他のプログラムモジュール37およびプログラムデータ38を含む、いくつかのプログラムモジュールを記憶することができる。ユーザは、キーボード40、ポインティングデバイス42などの入力装置を介して、コマンドおよび情報をパーソナルコンピュータ20に入力することができる。この他の入力装置（図示せず）には例えば、マイクロホン、ジョイスティック、ゲームパッド、衛星アンテナ、スキャナなどがある。これらの入力装置および他の入力装置はたい

てい、システムバスに結合されたシリアルポートインタフェース46を介して処理ユニット21に接続されるが、パラレルポート、ゲームポート、ユニバーサルシリアルバス（USB）、ネットワークインタフェースカードなどの他のインタフェースによって接続することもできる。さらに、モニタ47または他のタイプのディスプレイ装置が、ビデオアダプタ48などのインタフェースを介してシステムバス23に接続されている。モニタの他に、パーソナルコンピュータは一般に、スピーカ、プリンタなどの周辺出力装置（図示せず）を含む。

【0011】パーソナルコンピュータ20は、リモートコンピュータ49などの1台または数台のリモートコンピュータへの論理接続を使用したネットワーク化された環境で動作することができる。リモートコンピュータ49は、別のパーソナルコンピュータ、サーバ、ルータ、ネットワークPC、ピア装置または他の普通のネットワークノードとすることができる。図1にはメモリ記憶装置50だけしか示されていないが、リモートコンピュータ49は一般に、パーソナルコンピュータ20に関して先に説明した全ての要素または多くの要素を含む。図1に示した論理接続は、ローカルエリアネットワーク（LAN）51およびワイドエリアネットワーク（WAN）52を含む。このようなネットワーキング環境は、事業所、企業内コンピュータネットワーク、イントラネットおよびインターネットでよく普通に見られるものである。

【0012】LANネットワーキング環境で使用するとき、パーソナルコンピュータ20は、ネットワークインタフェースまたはアダプタ53を介してローカルネットワーク51に接続される。WANネットワーキング環境で使用するとき、パーソナルコンピュータ20は一般に

モデム54、またはWAN52を介した通信を確立するためのその他の手段を含む。モデム54は、内部モデムでもまたは外部モデムでもよく、シリアルポートインタフェース46を介してシステムバス23に接続される。ネットワーク化された環境では、パーソナルコンピュータ20に関して図示したプログラムモジュール、またはその一部分を、遠隔メモリ記憶装置に記憶することができる。示されたネットワーク接続は例示的なものであり、コンピュータ間に通信リンクを確立する他の手段を使用することができることを理解されたい。

【0013】特に指示しない限り以下の説明では、1台または数台のコンピュータによって実行されるアクト (act) およびオペレーションの記号的表現を参照して、本発明を説明する。時にコンピュータ実行アクトまたはオペレーションと呼ばれるこのようなアクトおよびオペレーションは、コンピュータの処理ユニットによる、構造化された形態のデータを表す電気信号の操作を含むことを理解されたい。この操作では、データを変換し、またはコンピュータのメモリシステム中の位置にデータを維持する。これによって、コンピュータのオペレーションは当業者によってよく理解されている方法で再構成され、または別な方法で変更される。データが維持されるデータ構造は、データのフォーマットによって定義された特定の特性を有するメモリの物理的位置である。しかし、本発明は以上の文脈で説明されるが、これは限定を意味しない。当業者なら分かつとおり、以降に説明するさまざまなアクトおよびオペレーションはハードウェアとして実装することもできるからである。

【0014】次に図2を参照する。本発明は、セキュリティ機構、特にケルベロス認証プロトコルを実装したセキュリティ機構を、セッションイニシエーションプロトコル(SIP)に基づくリクエストメッセージに組み込んで、SIPクライアント72とSIPプロキシサーバ74が相互に認証することができるようにし、かつシグナリングメッセージの完全性を保護するための方法を対象とする。SIPは、RFC(Request for Comments) 2543に定義されている。この文書は、その全体が参照によって本明細書に組み込まれる。

【0015】例えば、一般的なセッション開始オペレーションでは図2に示すように、別のユーザ80(例えば「ボブ(Bob)」)と話しをしたいSIPクライアント72(「呼出し元」)のユーザ76(例えば「アン(Ann)」)は、ボブがINVITEメッセージ82の意図された受信者であることを確認するINVITEメッセージを送る。このINVITEメッセージは、呼出し元SIPクライアントのドメインに対応するアウトバウンドプロキシサーバ74に送られる。図2に示すようにINVITEメッセージは、このシグナリングオペレーションに関与する複数のSIPプロキシを介して送

られ、その後、ボブのコンピュータ88のSIPクライアント86(呼出し先)に到達する。好ましい実施形態では、シグナリング経路上のSIPプロキシ間で転送中のSIPシグナリングメッセージのセキュリティが、IPsecプロトコルの下で、またはSSL(Secure Sockets Layer)プロトコルの下でパイプを通してメッセージを送ることによって保護される。この例ではSIPリクエストがINVITEリクエストだが、以下で説明する認証方式は、REGISTER、MESSAGE、SUBSCRIBE、SERVICEなどの他のタイプのSIPリクエストに対しても使用することができることを理解されたい。

【0016】シグナリングオペレーションのセキュリティおよびシグナリングメッセージの完全性を確保するため、アウトバウンドSIPプロキシサーバ74は、シグナリング経路90を介してINVITEメッセージ82を転送する前に、呼出し元SIPクライアント72のユーザ76の認証を求めることができる。次に図3を参照する。本発明によればプロキシサーバ74はINVITEメッセージに応答して、SIPクライアント72にチャレンジメッセージ96を送る。チャレンジメッセージ96は、まず最初にクライアント72がそのユーザをプロキシ74に対して認証しなければならないことを指示する、SIP仕様に定義されたステータスコード「407 Proxy Authentication Required(407プロキシ認証が必要)」を含む。SIP仕様によれば、チャレンジメッセージ96(以後「407メッセージ」と呼ぶ)は、認証のためにクライアントが使用しなければならないセキュリティ機構を指示するデータを含む「Proxy-Authenticate」ヘッダフィールド98を含む。Proxy-Authenticateヘッダの構文および内容ヘッダについては後に詳細に説明する。好ましい実施形態では、ケルベロスが好ましいセキュリティ機構だが、SIPフレームワークでは、NTLMプロトコルに基づくセキュリティ機構の使用も可能である。以下の説明では、特に指示しない限り、ケルベロスプロトコルに基づくセキュリティ機構を使用するものとする。

【0017】さらに図3を参照する。INVITEメッセージ82に回答したプロキシサーバ74から407メッセージ96を受け取ると、Proxy-Authenticateヘッダ98からSIPクライアント72は、ケルベロス機構を使用したユーザの認証をプロキシサーバが求めていると判断する。サーバチケットをまだ取得していない場合、クライアント72は次いで、SIPプロキシサーバ74のケルベロス・キー・ディストリビューション(配布)・センター(KDC: Key Distribution Center)100からサーバチケット108を取得する。一実施態様では、KDC100が、プロキシサーバ74のドメインコントロー



ラ102の一部である。サーバチケット108を取得した後、クライアント72は別のINVITEメッセージ110を送る。ただしこのときには、INVITEメッセージ110が、SIP仕様に基づくProxy-Authorization（プロキシ許可）ヘッダフィールド112を含んでいる。Proxy-Authorizationヘッダフィールド112は、プロキシサーバにアクセスするためのサーバチケット108を含む。サーバチケット108は、使用するセッションキー116を含んでいる。Proxy-Authorizationヘッダフィールドの構文および内容については後に詳細に説明する。任意選択で、Proxy-Authorizationヘッダにさらに、クライアント72に対して自体を認証するようプロキシサーバ74に求める、相互認証のためのリクエストを含めることもできる。

【0018】SIPプロキシサーバ74は、ケルベロスサーバチケットが埋め込まれた再送INVITEメッセージ110を受け取ると、サーバチケットを抜き出し、チケットの有効性を、KDC100と共有しているその長期キーでチケットを解読することによって検証する。チケットが有効である場合、ユーザ76は認証され、SIPプロキシサーバ74は、シグナリング経路上の次のプロキシ120にINVITEメッセージ110を転送する。クライアント72が、INVITEメッセージ110のProxy-Authorizationヘッダ112の中で相互認証を要求してきた場合、プロキシサーバ74は、ケルベロスサーバチケットに関連付けられたセッションキーを使用して、自体からクライアントへの将来のチケットに署名する。このメッセージは、クライアント72がプロキシ74を認証することを可能にするプロキシ74の証明書（credential）を含むProxy-Authentication Informationヘッダ124を含む。

【0019】最終的に、INVITEメッセージ110は呼出し先、すなわちボブのコンピュータ88のSIPクライアント86に届く。この呼勧誘を受け入れる場合、呼出し先は「200 OK」メッセージ126を返し、このメッセージは呼出し元へ送られる。呼接続が確立されると、呼出し元は、このシグナリング段階に関与したSIPプロキシを通すことなく、呼出し先と直接に通信することができる。

【0020】次に図4を参照する。本発明によれば、SIPクライアント72とSIPプロキシサーバ74の間で認証セキュリティアソシエーション（SA：security association）を確立するオペレーションは、状態機械128として見る事ができる。図4に示す実施形態では、好ましいセキュリティ機構がケルベロスであるが、任意選択でNTLMセキュリティ機構も使用することができ、この状態機械図は、この任

意選択の包含を反映したものになっている。

【0021】図4では、状態が円の中に、状態に関連して実行されるオペレーションが長方形のブロックの中に示されている。図4に示すように、状態機械の1つの状態が、セキュリティSAが確立されていない「SECURITY\_STATE\_NONE」状態132である。クライアント72が送ったINVITEメッセージに回答したプロキシ74から407チャレンジメッセージを受け取り、またはプロキシとプレ認証（pre-authentication）を実行すると決定すると、クライアント72は「SECURITY\_STATE\_ACQUIRING\_SA」状態136に入り、認証に必要なセキュリティアソシエーションデータを獲得する。認証に必要なセキュリティアソシエーションデータは、選択したセキュリティ機構によって異なる。

【0022】セキュリティアソシエーションは一般に、クライアントとSIPプロキシが共有の秘密を安全な方法で交換し、この秘密を使用して認証し、クライアントとプロキシによって交換される以降のメッセージの完全性を保護することができる状態と定義される。セキュリティ機構がケルベロスである場合、セキュリティアソシエーションは、プロキシに対するケルベロスサーバチケットおよびセッションキーを含む。ケルベロスの場合、取得されたSAは完全である。すなわち、これを使用して、SIPクライアントのユーザをプロキシが十分に認証することができる。クライアントは次いで、このSA関連情報（例えばサーバの秘密を用いて暗号化されたケルベロスセッションキー）をプロキシに送る（段階138）。プロキシが、署名された200OKメッセージを返した場合（段階140）、認証は成功であり、セキュリティアソシエーションは確立されている。すなわちクライアントはSA\_Established状態142にある。しかし、プロキシが代わりに407チャレンジメッセージを返した場合（段階146）、クライアントは、プロキシが不良な状態にあり、そのためクライアントの正当な証明書を検証することができないでいるとみなす。次いでクライアントは「バックオフ」時間（例えば5分）のあいだ待機し（段階148）、その後再びSIPメッセージを送る。

【0023】SA\_Established状態142に入った後は、セキュリティアソシエーションが期限切れにならない限り、クライアントは再び認証を実行することなくプロキシに別のメッセージを送ることができる。しかしプロキシが407チャレンジメッセージを送った場合（段階150）、クライアントは、確立されたセキュリティアソシエーションをプロキシが何かの理由で取り下げたとみなす。その結果、クライアントは、SA\_Dropped状態156に入り、SECURITY\_STATE\_ACQUIRING\_SA状態136へ移って、プロキシとの認証を再び実施するための新し

いSAを獲得する。

【0024】先に述べたとおり、ユーザの認証に対して任意選択でNTLM機構を選択することができる。NTLMの状態移行はケルベロスの状態移行と概ね同じだが、NTLMは、始めに不完全なSAだけを獲得し（段階158）、不完全なSAを第1のメッセージに入れてプロキシに送るという違いがある。具体的には、NTLMの場合、SA関連情報を含むクライアントからの第1のリクエストは、クライアントのセキュリティ関連機能（例えばクライアントがサポートするプロトコルのバージョン、サポートする署名アルゴリズムなど）を伝える。これに回答してプロキシサーバは、そのNTLM関連機能および一般に「nonce」と呼ばれるランダムなバイト列を含む自体の認証データを含んだ第2の407チャレンジを送る（段階160）。これに回答してクライアントは、それ自体の名称およびプロキシによって送られた「nonce」値のハッシュにその証明書を使用して署名する。これは、NTLMインプリメンテーションによって内部的に処理される。プロキシサーバは、\*

```
Proxy-Authenticate = "Proxy-Authenticate" scheme kerb-challenge gssapi-data
Scheme               = "kerberos" | "NTLM" | "Negotiate"
kerb-challenge        = 1# (realm | targetname | [opaque] | qop-options | gssapi-data)
targetname            = "targetname" "=" <"> URI (1*SP URI) <">
URI                   = absoluteURI | abs_path
opaque                = "opaque" "=" quoted-string
qop-options           = "qop" "=" <"> 1#qop-value <">
qop-value             = "auth" | "auth-int" | token
gssapi-data           = "gssapi-data" "=" (token | quoted-string)
```

【0028】ここに記載したProxy-Authenticateヘッダの構文は、「HTTP Authentication: Basic and Digest Access Authentication」という表題のIETF RFC 2617に定義されている「WWW-Authenticate Response Header」と同様である。この文書はその全体が参照によって本明細書に組み込まれる。任意選択のパラメータ「algorithm」および「stale」は省かれており、ヘッダの「scheme」フィールドは、サーバに対して自体を認証するのに、サーバによって提案された認証機構のうちどの認証機構を使用したいかをクライアントが選択することを可能にする。クライアントがケルベロス機構をサポートしている場合には、クライアントはケルベロス機構を選択することが好ましく、サポートしていない場合にはNTLM認証機構を選択する。

【0029】realmパラメータは、SIPプロキシが属し、クライアントがアクセスしようとしているSI

\* クライアントの認証データを検証し、ドメインコントローラの助けを借りてセッションキーを得る。SIPプロキシが意図された受信者でない場合、プロキシは、シグナリング経路上の次のホップにSIPリクエストを転送し、送信側SIPクライアントへの次のメッセージ（例えば受信者からの200OKメッセージ）に署名する（段階140）。

【0025】SIPクライアントとSIPプロキシの間の認証目的のメッセージ交換に関与するさまざまなSIPヘッダの構文については後に説明する。

【0026】（407応答）先に述べたとおり、INVITEメッセージを送ったSIPクライアント（またはそのユーザ）の識別を要求したい場合、SIPプロキシサーバ74は、Proxy-Authenticateヘッダを含む407メッセージをクライアントに送る。認証のためにケルベロスセキュリティ機構の使用を必要とする好ましい実施形態のProxy-Authenticateヘッダの構文は以下のとおりである。

【0027】

Pサービスプロバイダの固有の識別子である。認証のためにユーザが提供する必要がある正しい証明書セットをユーザが識別するのを助けるため、realm列はユーザに表示される。「targetname」パラメータは常に必要なパラメータであり、SIPプロキシのFQDNを伝達するのに使用される。このパラメータの実際の内容は、クライアントがSAを確立しているプロキシを追跡するのを助ける。それは、応答が自体に対するものなのか、または他のプロキシに対するものなのかをプロキシが判定するのを助ける。「opaque」パラメータは、サーバが、確立されている特定のSAに索引をつけるのに使用され、後に説明するように、クライアントがSAに対して生成する将来のProxy-Authenticationヘッダ中に反映されなければならない。

【0030】この実施形態では、IETF RFC 2078（その全体が参照によって本明細書に組み込まれる）に定義されたGSS-API（Generic Security Service Applicati

on Programming Interface) が実装されており、通信しているアプリケーション間でメッセージを安全に交換するのに使用されるとみなす。GSS-APIは特に、通信アプリケーションが、別のアプリケーションに関連付けられたユーザを認証することを可能にする。Proxy-Authenticateヘッダおよび後述するProxy-Authorizationヘッダのgssapi-dataフィールドは、NTLMおよびケルベロスセキュリティパッケージを実装したセキュリティAPIによって、SAネゴシエーション段階の間に返されたデータを保持するためのものである。これらのAPIは、クライアントからプロキシへ、およびプロキシからクライアントへ送る必要があるgssapiデータを返す。gssapiデータは、SIPクライアント/プロキシインプリメンテーションにとって不透明であり、セキュリティAPIだけが解釈することができる。qopパラメータは、クライアントが従って欲しいとサーバが欲しているセキュリティのレベルをクライアントに教える。qopパラメータ値は常に、この機構によって提供されるセキュリティレベルがユーザの認証であることを指示する「auth」にセットされている。

【0031】Proxy-Authenticateヘッダフィールドの一例を以下に示す。

【0032】

Proxy-Authenticate: Negotiate

realm = "Microsoft RTC Service provider",

opaque = "ABCDEF456789"

\*

Proxy-Authorization = "Proxy-Authorization" ":" scheme kerbresponse realm

m

message-qop targetname

kerb-response = l# ( [crand] | [response] | [opaque] | [gssapi-data] )

message-qop = "qop" "=" qop-value

crand = "crand" "=" crand-value

crand-value = crand-value

response = "response" "=" request-digest

request-digest = <"> 32LHEX <">

LHEX = "0" | "1" | "2" | "3" |

"4" | "5" | "6" | "7" |

"8" | "9" | "a" | "b" |

"c" | "d" | "e" | "f" |

【0036】ここに記載したProxy-Authorizationヘッダの構文は、IETF RFC 2617に定義されている「Authorization Request Header」と同様である。ただし、任意選択のパラメータ「algorithm」および「URI」は省かれている。Proxy-Authorizationヘッダは、リクエストURIおよびViaヘッダの後に追加される。署名は、セッションキー

\* qop = "auth",

gssapi-data = "ABCD345678yuikjhlbcdfsaqwety"

【0033】許可されたクライアントだけを許すとされ、クライアントから着信するSIPパケットが署名を含まない場合には一般に、SIPプロキシは、SIPクライアントの識別を要求するであろう。(リポートなどのため)SIPプロキシがこのSIP URIに対するセキュリティアソシエーションを失った場合にも、SIPプロキシはクライアントの認証を要求するであろう。クライアントが使用している許可パラメータとSIPプロキシが期待しているものの間にミスマッチがある場合、SIPプロキシは、クライアントが従って欲しいとSIPプロキシが欲している正確な許可パラメータを運んでいる407メッセージを使用して、クライアントの認証を要求するであろう。

【0034】(407チャレンジに対するクライアントの応答)407チャレンジに応答してSIPクライアントは、407チャレンジメッセージを介してSIPプロキシから送られた認証パラメータに従った署名を生成しようとする。SIPクライアントはCseq値を増分し、チャレンジの対象となった最初のSIPリクエストを、Proxy-Authorizationリクエストヘッダの中に入れられた許可情報とともに再送する。好ましい実施形態におけるProxy-Authorizationリクエストヘッダの構文は以下のとおりである。

【0035】

を使用して、以下のフィールドにわたって計算される。

【0037】

—FromヘッダURI

—ToヘッダURI

—Fromヘッダタグ

—Toヘッダタグ

—Proxy-Authorizationヘッダ中の「crand」パラメータ、またはProxy-Aut

hentication-Infoヘッダ中の「srand」パラメータ

-SIPメッセージExpiresヘッダ中のExpires値

署名には、SIPメッセージのメッセージ本体は含まれていない。proxy-authorizationヘッダは、gasapi-dataパラメータまたはresponse（署名）パラメータを含む。

【0038】以下に、407チャレンジに対するクライアントの応答におけるProxy-Authenticationヘッダの例を示す。

【0039】

```
Proxy-Authorization: Negotiate
realm = "Microsoft RTC Service Provider",
response = "ABCD87654cvx",
opaque = "ABCD1234",
crand = "1234"
qop = "auth"
targetname = "server1.domainA.microsoft.com"
または
Proxy-Authorization: Negotiate
realm = "Microsoft RTC Service Provider",
opaque = "ABCD1234",
gssapi-data = "ABCDEF123456",
qop = "auth",
```

\*

```
ProxyAuthenticationInfo = "Proxy-Authentication-Info" ":" auth-info
auth-info = 1# (message-qop | response-auth | srand)
response-auth = "rspauth" "=" response-digest
response-digest = <"> *LHEX <">
srand = "srand" "=" srand-value
srand-value = quoted-string
```

【0043】Proxy-Authentication-Infoヘッダ中の「rspauth」パラメータは、この応答に対する（認証を求めているプロキシの）署名を運ぶ。「srand」パラメータは、SA確立段階後にサーバが、クライアントに送るメッセージに署名するのに使用される。このパラメータは、サーバによって生成されるランダムな文字列であり、生成されたメッセージのハッシュ／署名にランダム性の要素を導入するのに使用される。

【0044】以下に、Proxy-Authentication-Informationヘッダの一例を示す。

【0045】

```
Proxy-Authentication-Info: Negotiate
realm = "Microsoft RTC Service Provider",
qop = "auth",
rspauth = "ABCD87654cvx",
srand = "9876543210",
targetuame = "server1.domainA.microsoft.com"
```

\*targetname = "server1.domainA.microsoft.com"

【0040】プロキシからの407チャレンジに応答するときの他に、クライアントは、自体を初めてSIPプロキシに登録するときにもこのヘッダを送る。SIPクライアントが自体をプロキシサーバに登録し、セッションのためにセキュリティアソシエーションを初期化するプロセスにあるとき、Proxy-Authenticationヘッダは、「gssapi-data」パラメータを含む。

【0041】（相互認証）ある種のシナリオでは、SIPプロキシとSIPクライアントの間で相互認証を確立することが必要である。クライアントは、特定のプロキシサーバに対してクライアントが有する準備プロファイルから、相互認証が必要か否かを判断する。相互認証が使用可能な場合、クライアントは、GSSAPIの標準バージョンを使用して相互認証のためのセキュリティアソシエーションを初期化する。相互認証が使用可能な場合にはさらに、サーバが、SIPクライアントに送る全てのパケットに署名する必要がある。この署名は、Proxy-Authentication-Informationリクエストヘッダに入れて運ばれる。Proxy-Authenticate-Informationの構文は以下のとおりである。

【0042】

\*

【0046】SIPフレームワークでは一般に、REGISTERリクエストを使用した登録プロセスの間に、SIPクライアントが、SIPプロキシとのセキュリティアソシエーションを確立することができる。登録によってSIPクライアントは、SIPプロキシからメッセージを受け取ることができるようになる。SIPクライアントが自体をSIPプロキシに登録するときには、SIPクライアントは同時に、ケルベロスチケットなどの認証データをREGISTERメッセージに入れて送ることによって、自体のユーザをSIPプロキシサーバに対して認証することができる。SIPクライアントがSIPプロキシにすでに登録されており、かつSIPプロキシに対して認証されている場合、INVITEなどのSIPリクエストをクライアントが送るときには、クライアントからのリクエストメッセージが、SA確立プロセスの間に交換されたケルベロスセッションキーを使用して署名される。

【0047】とは言え、自体をサーバに登録しなくても、SIPクライアントはリクエストメッセージをSIP

Pプロキシに送ることができる。呼出し元がプロキシに対して認証されていない場合には(たとえSIPクライアントがプロキシに登録されている場合であっても)、SIPプロキシはそのリクエストを次のホップに転送しない。その代わりにプロキシは、SIPクライアントにチャレンジを送る。

【0048】このチャレンジは、クライアントがこのSIPサーバと、セキュリティアソシエーションを確立する必要があることを指示する。クライアントは、セキュリティアソシエーションデータを含むリクエストを再び送ることによってSAを確立することができ、あるいは、登録はすでにされているが、SAがまだ確立されていない場合には、このサーバへの自体の登録をリフレッシュすることによってSAを確立することができる。登録リフレッシュを使用してSAを確立し、次いで有効な署名を有するSIPリクエストを送ることには、登録が良好な状態にあることが保証されるという利点がある。

【0049】さらに、SIPクライアントがSIPプロキシに対する登録を抹消するたびに、SIPクライアントとSIPプロキシの間のセキュリティアソシエーション(SA)は失われ、新しいセキュリティアソシエーションを再び交渉し直さなければならない。さらに、SIPクライアントの登録が失効すると、プロキシサーバは、対応するセキュリティコンテキストをSAのリストから削除する。登録をリフレッシュするたびに、SIPクライアントは、認証セキュリティアソシエーションもリフレッシュしなければならない。

【0050】ケルベロスプロトコルに基づくセキュリティ機構を使用する好ましい実施形態では、送信側SIPクライアントのユーザの認証がSIPプロキシ/レジストラによって要求された場合に、SIPクライアントがSIPプロキシに登録するたびに、ケルベロスキーディストリビューションセンター(KDC)からのケルベロスチケットが要求される。ケルベロスチケットを受け取ると、SIPクライアントはこのチケットを解読する。解読されたチケットは、セッションキーおよびこのケルベロスセッションの他のいくつかの特性を含む。このチケットはさらに、サーバの証明書で暗号化されたセッションキーおよび他のセッション関連パラメータを含む。この部分は、gssapi-dataフィールドの、pOutputパラメータに入れて返され、re-INVITEリクエストに入れてプロキシに送られる。

【0051】SIPフレームワーク内でのセキュリティ機構のオペレーションの明瞭な理解を容易にするため、クライアント・プロキシのケルベロス認証の具体的な例を図2を参照して説明する。この例では、SIPプロキシサーバ74が、ドメイン「domainA.Microsoft.com:Server1」のKDC170との共有秘密鍵をすでに生成していると仮定する。この例では「Server1」が、SIPプロキシ/レ

ジストラのコード名として使用される。KDC170は、プロキシサーバ74がserver\_ID=server1.domainB.microsoft.comであると知っている。プロキシサーバ74はさらに、証明書ハンドルを獲得して、クライアントから来る認証リクエストに応答する準備を整える。サーバ証明書は、サーバ認証または相互認証をサポートしているセキュリティプロトコルにおいてSIPクライアント74に対してプロキシサーバ74を認証するのに使用される。プロキシサーバ74は、このサーバを起動するのに使用されるサービスアカウントによって定義されるその証明書に対するハンドルを取得する。サーバはこのハンドルを、SSPI(Security Support Provider Interface)の関数AcquireCredentialsHandleを呼び出すことによって取得する。

【0052】図2の例では、SIPクライアント72のユーザ76がアンである。アンは、NTドメインにアカウントを有し、一日の始まりに、以下の情報を使用して自分のアカウントにログオンする。

【0053】

```
UserID / principal name = ann@microsoft.com
Preferred_email = ann@microsoft.com
User_domain = domainA.Microsoft.com
Workstation = ann1.domainA.Microsoft.com
```

【0054】ボブと通話したいとき、アンは、自分のワークステーション78上のSIPクライアント72を起動する(SIPクライアントはサービスとして自動的に起動することができる。ただしユーザのセキュリティコンテキストで走らなければならない)。SIPクライアント72は、DNSを使用してアウトバウンドプロキシサーバ74を見つける。この例で使用するアウトバウンドプロキシサーバ74は、Server1.domainB.Microsoft.comとして識別される。アンは、自分がbob@microsoft.comと通話したいことを指示する。アンのSIPクライアント72は次いで、Server1.domainB.Microsoft.comにINVITEメッセージ82を送る。INVITEメッセージは以下の情報を含む。

【0055】

```
INVITE bob@microsoft.com
From: ann@microsoft.com
To: bob@microsoft.com
```

【0056】この例の説明を簡潔かつ明瞭に保つため、このINVITEメッセージまたはシグナリング処理で交換される他のメッセージに含まれる全てのデータを示すわけではない。SIPプロキシサーバ74は、Microsoft.comのユーザ名空間に対してなされた呼に対する全てのINVITEリクエストが認証されることを必要とするように構成されている。その結果、S

IPプロキシサーバ74はINVTリクエストに  
答して、ケルベロスを使用してユーザ（アン）を認証す  
るようSIPクライアント74に求める407メッセー\*

Proxy-Authenticate: Kerberos realm = domainB.microsoft.com

targetname = "server1.domainA.Microsoft.com" opaque = "someopaquedata"

【0058】opaque値は、この呼に対して使用す  
るセキュリティコンテキストを識別するためにプロキシ  
によって初期化される。そのために、プロキシサーバ7  
4はこのとき、関数AcceptSecurityCo  
ncontextを呼び出し、pOutputをbase64 10  
でコード化した結果をopaqueに入れて返す。クラ  
イアントおよびサーバは、このopaque値を使用し  
て、認証の継続、またはAuthorizationリ  
クエストヘッダを使用した同じサーバへの以降のリク  
エストの再認証のため特定のサーバに対して使用するセ  
キュリティコンテキストを識別する。

【0059】アンのワークステーション上のSIPクラ  
イアント72は、認証が必要であることを指示する40  
7メッセージ96を受け取ると、Server1.do  
mainB.Microsoft.comと通信するた  
めの有効なセッションキーを持っているかどうかをチェ  
ックする。まだ所有していない場合には、このドメイン  
内のKDCと接触して、アウトバウンドSIPプロキシ  
にアクセスするためのセッションキーを取得する必要  
がある。この例では、407メッセージの中に指定され  
たrealmからクライアントは、自体のドメインとは  
異なるドメインにプロキシがあることを知る。

【0060】プロキシサーバ74への安全な接続を確立  
するため、クライアント72は、認証リクエストをプロ  
キシに送る前にアウトバウンド証明書ハンドルを獲得す  
る。これは、関数SSPIを呼び出すことによって実行  
される。SSPIは、ネットワーク化されたアプリケー  
ションが、いくつかあるうちの1つのセキュリティサポ  
ートプロバイダ（SSP）にコールして、認証済みの接  
続を確立し、確立された接続を介してデータを安全に交  
換する手段を提供する。認証セットアップに関与する2  
つのクライアント側SSPI関数がある。AcquireCred  
entialsHandle関数は、以前に  
取得したログオン証明書への参照を取得する。関数In  
italizeSecurityContextは、  
最初の認証リクエストセキュリティトークンを生成す  
る。InitializeSecurityConte  
xtを呼び出すと、407メッセージから取得したop  
aque値がpInputに入れて渡される。クライ  
アントは、この関数のtfContextReqパラメ  
ータをリクエストMUTUAL\_AUTHにセットする。  
pfContextAttrポインタは、mutual  
-authが「リクエスト」されたことをケルベロスモ  
ジュール180がクライアントに知らせる方法である。  
この情報は、クライアントのケルベロスモジュール18

\*ジ96を送る。407メッセージは以下のデータを含  
む。

【0057】

0によって生み出されるKERB\_\_AS\_\_REQの一部  
であり、クライアントが相互認証を求めていることをサ  
ーバ（ここではSIPプロキシ）に知らせるsecBu  
ffer（pOutput）に入れて渡される。これは  
KERBリクエストの一部なので、SIP機構（ヘッダ  
／パラメータ）が相互認証を要求する必要はない。

【0061】図2に示した例では、API関数Init  
ializeSecurityContextを呼び出  
すことによって、以下のケルベロス論理が生じる。最初  
にクライアント72が、DomainB（ドメインB）  
のプロキシサーバ74へのサーバチケットをクライアン  
ト72に与えるよう、domainA.Microsoft.  
comドメインのKDC170に求める。dom  
ainA.Microsoft.comのKDC170  
はクライアント72に、corp.Microsoft.  
comのKDC172への照会チケットを送る。こ  
の照会チケットは、この2つのKDCによって共有され  
たドメイン間キーの中に暗号化されている。この照会チ  
ケットを使用して、クライアントは、DomainBに  
あるサーバへのサーバチケットをクライアントに与える  
よう、corp.Microsoft.comのKDC  
172に求める。

【0062】これに回答してKDC172は、Doma  
inBのKDC176への照会チケットをクライアント  
に送る。このチケットは、KDC172がDomain  
BのKDC176と共有するドメイン間キーの中に暗号  
化されている。クライアントは次いで、DomainB  
にあるプロキシサーバ74へのチケットをクライアント  
に与えるよう、DomainBのKDC176に求め  
る。KDC176は、プロキシサーバ74にアクセスす  
るためのサーバチケット108をクライアントに送る。  
KDC176は、アンのログオンセッションキーを用い  
てこのセッションキーの1つのコピーを暗号化し、セ  
ッションキーの別のコピーをアンの許可データとともにサ  
ーバチケットに埋め込み、プロキシサーバの長期キーを  
用いてサーバチケットを暗号化する。KDC176は次  
いで、これらの証明書を、Kerberos Tick  
et-Granting Service Reply  
（KRB\_\_TGS\_\_REP）に入れてクライアント72  
に送る。

【0063】このように、InitializeSec  
urityContextを呼び出すことによって、ク  
ライアントマシンのケルベロスモジュール180がKD  
CとのTGS交換を開始する。この交換によって返され  
る値は、プロキシに送るメッセージに署名するためのセ

ッションキーである。

【0064】その後、SIPクライアント72は、SIPプロキシに送る新しいINVITEメッセージ110（「re-INVITE」メッセージとも呼ばれる）を生成する。この新しいINVITEメッセージ110は、クライアントがKDC176から受け取ったサーバチケットを含むGSS-APIデータをその中に含ん \*

```
INVITE bob@microsoft.com
From: ann@microsoft.com
To: bob@microsoft.com
Proxy-authorization: gss-scheme opaque gssapi-rdata
Opaque = someopaquedata
Gssapi-rdata = base64 {pOutput} = session key to the proxy
```

【0066】このINVITEメッセージは、プロキシサーバに対してKRB\_\_AP\_\_REQと等価の内容を実行する。

【0067】メッセージの完全性を保護し、自体を認証する（すなわちメッセージの出処を証明する）ため、クライアントは、セッションキーを用いてINVITEメッセージ110に署名する。さもないと、第三者がこのINVITEメッセージをかぎつけ、OpaqueおよびGssapi-rdata値を手に入れ、偽のINVITEメッセージを同じサーバに送って、この第三者とそれが選択した任意の転送先との間で通話をおこなう可能性がある。このことは、サーバへのセッションキーが有効である間（デフォルトでは8時間）、クライアントの認証を「盗む」ことができることを意味する。INVITEメッセージに署名することで、第三者がOpaqueおよびGssapi-rdataを取り出すことを防ぐことはできないが、新しいINVITEメッセージを生成して好き勝手に通話することは防ぐことができる。この問題を回避するためには、サーバが、署名されたリクエストしか受け入れないように構成されていなければならない。

【0068】クライアント72は、MakeSignature APIを使用し、これを呼び出して、（407メッセージのopaque中に識別された）この呼出しで使用されるセキュリティコンテキストにphContextをセットし、その内容を渡してpMessageに署名する。この呼出しの出力は、pMessageの中に返された署名されたメッセージである。クライアントは、この署名をINVITE110に追加する。再送されたINVITEメッセージ110を受け取ると、プロキシサーバ74は、Proxy-Authorizationヘッダの中のopaque値をチェックし、所与のphContext値（所与のセキュリティコンテキストへのハンドル）と相関させる。プロキシサーバ74は、gssapi-rdataを取り出し、AcceptSecurityContextAPI関数を呼び出し、proxy-authorizationヘッ

\*だ、先に説明したproxy-authorizationヘッダを含む。セッションキーは、InitializeSecurityContext呼出しによって返されたpOutputバッファの中に入れて返された値である。したがって、新しいINVITEメッセージ110は以下のデータを含む。

【0065】

ダから得たgssapi-rdata値をこのAPI関数のpInput成分の中に渡すことによって、これを自体のケルベロスモジュール182を渡す。ケルベロスモジュール182は、プロキシの長期キーを使用してサーバチケットを解読し、アンの許可データおよびセッションキーを抜き出す。ケルベロスモジュール182は、セッションキーを使用してアンのオーセンティケータ（authenticator）を解読し、次いで中のタイムスタンプを評価する。

【0069】オーセンティケータがこのテストにパスした場合、ケルベロスモジュール182は、クライアントのリクエストの中の相互認証フラグを探す。フラグがセットされている場合、ケルベロスモジュール182はセッションキーを使用して、アンのオーセンティケータからの時刻を暗号化し、その結果を、Kerberos Application Reply（KRB\_\_AP\_\_REP）の中に返す。これによって、AcceptSecurityContextが呼び出され、SEC\_\_E\_\_OK戻り値が返され、オーセンティケータはpOutputバッファを使用してこのAPIを通過する。ユーザが認証されると、SIPプロキシ/レジストラはこのリクエストを処理し、INVITEメッセージを、SIPシグナリング経路上の次のホップに転送する。

【0070】プロキシのSIPコンポーネントは次いで、SIPクライアントに転送する次のメッセージを使用してプロキシのオーセンティケータをクライアントに渡し、これによってクライアントがサーバを認証できるようにする。示された例では、このメッセージが「200OK」メッセージである。このメッセージは、SIPプロキシによって生成されたものではない。この200応答は、INVITEリクエストに応答して呼出し先が生成する。SIPプロキシは単に、呼出し元に転送する前にセッションキーを用いてこの応答に署名するだけである。

【0071】先に説明したとおり、オーセンティケータは、Proxy-Authentication-Informationヘッダの中にある。このヘッダはさ

らに、クライアントが、この応答を正しいセキュリティコンテキストと一致させるためのopaque値を含む。

【0072】アンのワークステーション上のSIPクライアント72は、「200OK」メッセージを受け取ると、Proxy-Authentication-Informationヘッダを抜き出し、InitializeSecurityContextを呼び出す。pHContext値はopaqueの中の値にセットされ、pInputバッファはresponse-digestにセットされる。クライアント上のケルベロスモジュール180は、プロキシと共有するセッションキーを用いてプロキシのオーセンティケータを解読し、プロキシによって返された時刻を、クライアントのオリジナルのオーセンティケータの中の時刻と比較する。2つの時刻が一致した場合、このInitializeSecurityContextの呼出しはSEC\_E\_OKを返し、クライアントは、プロキシが本物であることを知る。一致しない場合にはクライアントはこの呼を取りやめなければならない。クライアントは、要求したことをサーバが実行すると信じることができないので、CANCELを送ってこの呼を強制終了せざるを得ない。

【0073】上に説明した例では、認証が、まず最初にSIPクライアントが認証データを含まないINVITEを送り、次いで、認証が必要であることを指示するプロキシからの407メッセージに回答して認証データを別のINVITEに入れて送るシナリオで実施される。その代わりにクライアントは、プロキシに送る最初のINVITEに必要な認証データを含めることもできる。そのためにクライアント72は、SIPの下で通話するためにユーザが使用する前に、プロキシに対するサーバチケットをKDC176から取得する。次いで必要な認証データを、先に説明したProxy-Authorizationリクエストヘッダに入れる。こうすることによって、プロキシがクライアントに407チャレンジを送って認証データを要求する必要がなくなる。さらに、先に説明した認証オペレーションの例ではSIPプロキシが1つしか関与していないが、呼出し元と呼出し先の間のSIPシグナリング経路上には一般に複数のS\*

\*IPプロキシがあり、2つ以上のSIPプロキシが、呼出し元クライアントの認証を要求する可能性がある。例えば、図5に示す単純化されたケースでは、SIPクライアントのアウトバウンドプロキシサーバ74の他に別のSIPプロキシサーバ120があり、両方のプロキシが、INVITEメッセージを転送する前にクライアントの認証を必要とする。この場合には、クライアント72がまず、図4に関して先に説明した同じプロセスを経て、アウトバウンドSIPサーバ74に対して自体を認証する。クライアントを認証した後、プロキシサーバ74は第2のプロキシ120にINVITEを送り、第2のプロキシ120は次いで、クライアントに407チャレンジ190を送る。これに回答してクライアントは、第2のプロキシサーバ120用のケルベロスサーバチケットを含んだProxy-Authorizationヘッダを有する別の新しいINVITE192を送る。クライアントを認証した後、第2のプロキシは呼出し先にINVITE192を渡す。

【0074】以下の説明は、ケルベロスまたはNTLMセキュリティ機構に基づいて認証を実行するさまざまなメッセージフローのシナリオにおいて、Proxy Authenticate、Proxy AuthorizationおよびProxy-Authentication-Informationヘッダをどのように使用するかを説明する追加の例を提供する。図6を参照する。このケースでは、SIPクライアント72が、体をプロキシサーバに登録するときにケルベロススペースのプレ認証を実行する。クライアントは、プロキシに対するケルベロスサーバチケットを含むProxy-Authorizationヘッダを含むREGISTERリクエスト200、および先に説明した相互認証のためのリクエストを送る。サーバチケットに基づいてクライアントを認証した後、プロキシは、クライアントが使用してプロキシを認証することができるプロキシ認証データを含むProxy-Authentication-Informationヘッダとともに、200OKメッセージ202を返す。REGISTERおよび200OKメッセージの内容の例を以下に示す。

【0075】

```
REGISTER sip: nickn@microsoft.com SIP/2.0
Via: SIP/2.0/UDP www.xxx.yyy.zzz:5060
From: "Nick North" <sip:nickn@microsoft.com>
To: "Mark Mars" <sip:markmm@microsoft.com>
Call-ID: 123456789@microsoft.com
CSeq: 1 REGISTER
Contact: <sip:123.45.67.89:5060>
Proxy-Authorization: Negotiate
realm = "Microsoft RTC Service Provider", qop = "auth", gssapdata = "34
fcbaed78902QWERTY", targetname =
"server1 doaminA.microsoft.com"
```



User-Agent: Microsoft-RTC/1.0  
 Content-Length: 0  
 SIP/2.0 200 OK  
 Via: SIP/2.0/UDP www.xxx.yyy.zzz:5060  
 Proxy-Authentication-Info: Negotiate qop = auth, rspauth = "ABCD87564cvx",  
 srand = "9876543210" realm = "Microsoft RTC Service Provider" targetname =  
 "server1.doaminA.microsoft.com"  
 From: "Nick North" <sip:nickn@microsoft.com>  
 To: "Mark Mars" <sip:markm@microsoft.com>  
 Call-ID: 123456789@ms.com  
 CSeq: 1 REGISTER  
 Contact: "Nick north" <sip:www.xxx.yyy.zzz>  
 User-Agent: Microsoft-RTC/1.0  
 Content-Length: 0

【0076】図7に、ケルベロスベースのチャレンジド認証(challenged authentication)のシナリオを示す。この例では、クライアント72がまず、Proxy-Authorization 20  
 情報を含まないINVITE206をプロキシ74に送る。プロキシはこれに応答して、認証が必要であることを指示するProxy-Authenticateヘッダを含む407メッセージ208を送る。この407メッセージに  
 応答してクライアントは、必要なケルベロス認証データを含むProxy-Authorizationヘッダを有するREGISTERリクエスト210\*

\*を送る。プロキシは、プロキシ自体についての認証情報を含むProxy Authentication Informationヘッダを有する「200OK」メッセージ212を返す。Proxy Authentication Informationヘッダの中のデータに基づいてプロキシを認証した後、クライアントは、Proxy-Authorizationヘッダを有する第2のINVITE214を送る。このプロセスの例示的なメッセージを以下に示す。

【0077】

SIP/2.0 407 Proxy Authorization Required  
 Via: SIP/2.0/UDP www.xxx.yyy.zzz:5060  
 From: "Nick North" <sip:nickn@microsoft.com>  
 To: "Mark Mars" <sip:markm@microsoft.com>  
 Call-ID: 12345600@PC1.ms.com  
 CSeq: 1 INVITE  
 Proxy-Authenticate: Negotiate realm = "Microsoft RTC Service Provider",  
 targetname = "server1.doaminA.microsoft.com", qop = "auth"  
 Contact: <sip:123.45.67.89:5060>  
 User-Agent: Microsoft-RTC/1.0  
 Content-Length: 0

REGISTER sip:nickn@microsoft.com SIP/2.0  
 Via: SIP/2.0/UDP www.xxx.yyy.zzz:5060  
 From: "Nick North" <sip:nickn@microsoft.com>  
 To: "Mark Mars" <sip:markm@microsoft.com>  
 Call-ID: 123456789@microsoft.com  
 CSeq: 1 REGISTER  
 Contact: <sip:123.45.67.89:5060>  
 Proxy-Authorization: Negotiate realm = "Microsoft RTC Service Provider",  
 opaque = "ABC01234", qop = "auth", gssapi-data = "34fcbaed78902QWERTY"

```

targetname = "server1.domainA.microsoft.com"
User-Agent: Microsoft-RTC/1.0
Content-Length: 0
SIP/2.0 200 OK
Via: SIP/2.0/UDP www.xxx.yyy.zzz:5060
Proxy-Authentication-Info: Negotiate qop= "auth", rspauth =
"ABCD87564cvx", srnd = "9876543210"
targetname = "server1.doaminA.microsoft.com" realm = "Microsoft RTC Serv
ice
Provider",
From = "Nick North" <sip: nickn@microsoft.com>
To: "Mark Mars" <sip:markm@microsoft.com>
Call-ID: 123456789@ms.com
CSeq: 1 REGISTER
Contact: <sip:123.45.67.89:5060>
User-Agent: Microsoft-RTC/1.0
Content-Length: 0

```

【0078】次に図8を参照する。先に述べたとおり、好ましい実施形態では任意選択で、クライアント・プロキシ認証にNTLMセキュリティ機構を使用することができる。この場合、クライアントがまず、認証データを含まないINVITEメッセージ220を送り、プロキシが407メッセージを返す。この407メッセージ222のProxy Authenticateヘッダは、認証に対してNTLMを使用しなければならないことを指示する。次いでクライアントが、NTLMプロトコルに基づくクライアントの認証データを含むProxy Authenticationヘッダを有するREGISTERメッセージ224を送る。

【0079】図4の状態機械に関連して先に述べたとおり、クライアントによって送られた認証データによって、プロキシはクライアントを認証することができるが、この認証データに基づいてセキュリティアソシエーションは完全には確立されない。そのためプロキシは、\*

\* Proxy Authenticateヘッダをやはり含む別の407チャレンジ226をクライアントに送る。クライアントは次いで、セキュリティアソシエーションを完成させるのに必要な認証データを含むProxy Authorizationヘッダを有する別のREGISTERリクエスト228を送る。プロキシサーバは、第2のREGISTERリクエストの中のデータに基づいてセキュリティアソシエーションを完成させ、プロキシについての認証データを含むProxy Authentication Informationヘッダを有する「200OK」メッセージ232を返す。「200OK」メッセージ232の中の認証データに基づいてクライアントはプロキシを認証し、次いで別のINVITEメッセージ236を送る。このプロセスの例示的なメッセージを以下に示す。

【0080】

```

SIP/2.0 407 Proxy Authorization Required
Via: SIP/2.0/UDP www.xxx.yyy.zzz:5060
From: "Nick North" <sip: nickn@microsoft.com>
To: "Mark Mars" <sip:markm@microsoft.com>
Call-ID: 12345600@PCL.ms.com
CSeq: 1 INVITE
Proxy-Authenticate: NTLM realm = "Microsoft RTC service Provider",
targetname = "server1.domainA.microsoft.com",
opaque = "ABCD1234", qop = "auth",
Contact: <sip:123.45.67.89:5060>
User-Agent: Microsoft-RTC/1.0
Content-Length: 0
REGISTER sip: nickn@microsoft.com SIP/2.0
Via: SIP/2.0/UDP www.xxx.yyy.zzz:5060
From: "Nick North" <sip: nickn@microsoft.com>
To: "Mark Mars" <sip:markm@microsoft.com>

```

33

34

Call-ID: 123456789@microsoft.com  
 CSeq: 1 REGISTER  
 Contact: <sip:123.45.67.89:5060>  
 Proxy-Authorization: NTLM realm = "Microsoft RTC Service Provider",  
 opaque = "ABCD1234", qop = "auth", gssapi-data = "34fcbaed78902QWERTY"  
 targetname = "server1.domainA.microsoft.com"  
 User-Agent: Microsoft-RTC/1.0  
 Content-Length: 0  
 SIP/2.0 407 Proxy Authorization Required  
 Via: SIP/2.0/UDP www.xxx.yyy.zzz:5060  
 From: "Nick North" <sip:nickn@microsoft.com>  
 To: "Mark Mars" <sip:markm@microsoft.com>  
 Call-ID: 12345600@PC1.ms.com  
 CSeq: 1 INVITE  
 Proxy-Authenticate: NTLM realm = "Microsoft RTC Service Provider",  
 targetname = "server1.domainA.microsoft.com", opaque="ABCD1234", qop = "  
 auth",  
 gssapi-data = "QWERTY789564NMJHKLasdcfg"  
 Contact: <sip:123.45.67.89:5060>  
 User-Agent: Microsoft-RTC/1.0  
 Content-Length: 0  
 REGISTER sip:nickn@microsoft.com SIP/2.0  
 Via: SIP/2.0/UDP www.xxx.yyy.zzz:5060  
 From: "Nick North" <sip:nickn@microsoft.com>  
 To: "Mark Mars" <sip:markm@microsoft.com>  
 Call-ID: 123456789@microsoft.com  
 CSeq: 2 REGISTER  
 Contact: <sip:123.45.67.89:5060>  
 Proxy-Authorization: NTLM realm = "Microsoft RTC Service Provider",  
 gssapi-data = "qqertyuioKMNFO09876" opaque = "ABCD1234", qop = "auth",  
 targetname = "server1.domainA.microsoft.com"  
 User-Agent: Microsoft-RTC/1.0  
 Content-Length: 0  
 SIP/2.0 200 OK  
 Via: SIP/2.0/UDP www.xxx.yyy.zzz:5060  
 Proxy-Authentication-Info: NTLM realm = "Microsoft RTC Service Prov  
 ider" qop = "auth",  
 rspauth = "ABCD87564cvx", srnd = "9876543210"  
 targetname = "server1.domainA.microsoft.com"  
 From: "Nick North" <sip:nickn@microsoft.com>  
 To: "Mark Mars" <sip:markm@microsoft.com>  
 Call-ID: 123456789@ms.com  
 CSeq: 2 REGISTER  
 Contact: <sip:123.45.67.89:5060>  
 User-Agent: Microsoft-RTC/1.0  
 Content-Length: 0  
 INVITE sip: markm@proxyl.wcom.com SIP/2.0  
 Via: SIP/2.0/UDP www.xxx.yyy.zzz:5060  
 Proxy-Authorization: NTLM realm = "Microsoft RTC Service Provider"

```

crand = "913082051",
response = 12345ABCDEF78909BCADE56", opaque = "ABCD1234", qop =
"auth", targetname = "server1.domainA.microsoft.com"
From: "Nick North" <sip:nickn@microsoft.com>
To: "Mark Mars" <sip:markm@microsoft.com>
Call-ID: 12345601@PC1.ms.com
CSeq: 2 INVITE
Contact: "Nick North" <sip:nickn@microsoft.com>
User-Agent: Microsoft-RTC/1.0
Content-Type: application/sdp
Content-Length: xxx

```

【0081】図9に、NTLMベースのプレ認証のシナリオを示す。このケースのメッセージフローは、ケルベロスベースのプレ認証のそれに似ているが、407チャレンジおよびREGISTERメッセージが追加されている点異なる。具体的にはクライアントが、NTLMを使用することを指示し、NTLM認証データを含むProxy Authorizationヘッダを含むREGISTERメッセージ240および相互認証のためのリクエストを送る。プロキシは、受け取ったNTLM認証データに基づいてクライアントを認証し、Proxy Authenticateヘッダを有する407チ\*

\*チャレンジ242を返す。クライアントは次いで、プロキシとのセキュリティアソシエーションを完成するためのNTLM認証データを含むProxy Authorizationヘッダを有する第2のREGISTERリクエスト244を送る。プロキシは次いで、Proxy Authentication Informationを有する「200OK」メッセージ246を返す。プロキシを認証した後、クライアントはプロキシに第2のINVITEメッセージ248を送る。このプロセスの例示的なメッセージを以下に示す。

【0082】

```

REGISTER sip:nickn@microsoft.com SIP/2.0
Via: SIP/2.0/UDP www.xxx.yyy.zzz:5060
From: "Nick North" <sip:nickn@microsoft.com>
To: "Mark Mars" <sip:markm@microsoft.com>
Call-ID: 123456789@microsoft.com
CSeq: 1 REGISTER
Contact: <sip:123.45.67.89:5060>
Proxy-Authorization: NTLM realm = "Microsoft RTC Service Provider",
opaque = "ABCD1234", qop = "auth", gssapi-data = "34fcbaed78902QWERTY",
targetname = "server1.dornainA.microsoft.com"
User-Agent: Microsoft-RTC/1.0
Content-Length: 0
SIP/2.0 407 Proxy Authorization Required
Via: SIP/2.0/UDP www.xxx.yyy.zzz:5060
From: "Nick North" <sip:nickn@microsoft.com>
To: "Mark Mars" <sip:markm@microsoft.com>
Call-ID: 12345600@PC1.ms.com
CSeq: 1 INVITE
Proxy-Authenticate: NTLM realm = "Microsoft RTC Service Provider",
targetname = "server1.domainA.microsoft.com", opaque = "ABCD1234",
qop = "auth",
gssapi-data = "QWERTY789564NMJHKLasdcfg",
targetname= "server1.domainA.microsoft.com"
Contact: <sip:123.45.67.89:5060>
User-Agent: Microsoft-RTC/1.0
Content-Length: 0
REGISTER sip:nickn@microsoft.com SIP/2.0

```

Via: SIP/2.0/UDP www.xxx.yyy.zzz:5060  
 From: "Nick North" <sip:nickn@microsoft.com>  
 To: "Mark Mars" <sip:markm@microsoft.com.>  
 Call-ID: 123456789@microsoft.com  
 CSeq: 2 REGISTER  
 Contact: <sip:123.45.67.89:5060>  
 Proxy-Authorization: NTLM realm = "Microsoft RTC Service Provider",  
 gssapi-data = "qqertyuioKMNFO09876" opaque = "ABCD1234", qop = "auth",  
 targetname = "server1.domainA.microsoft.com"  
 User-Agent: Microsoft-RTC/1.0  
 Content-Length: 0  
 SIP/2.0 200 OK  
 Via: SIP/2.0/UDP www.xxx.yyy.zzz:5060  
 Proxy-Authentication-Info: NTLM qop = "auth",  
 rspauth = "AECD87564cvx", srnd = "9876543210",  
 targetname = "server1.domainA.microsoft.com"  
 From: "Nick North" <sip:nickn@microsoft.com>  
 To: "Mark Mars" <sip:markm@microsoft.com>  
 Call-ID: 123456789@ms.com  
 CSeq: 2 REGISTER  
 Contact: <sip:123.45.67.89:5060>  
 User-Agent: Microsoft-RTC/1.0  
 Content-Length: 0  
 INVITE sip: markm@proxyl.wcom.com SIP/2.0  
 Via: SIP/2.0/UDP www.xxx.yyy.zzz:5060  
 Proxy-Authorization: NTLM realm = "Microsoft RTC Service Provider",  
 crand = "913082051",  
 response = "12345ABCDEF78909BCADE56", opaque = "ABCD1234", qop "auth" ta  
 rgetname = "server1.domainA.microsoft.com"  
 From: "Nick North" <sip:nickn@microsoft.com>  
 To: "Mark Mars" <sip:markm@microsoft.com>  
 Call-ID: 12345601@PCL.ms.com  
 CSeq: 2 INVITE  
 Contact: "Nick North" <sip:nickn@microsoft.com>  
 User-Agent: Microsoft-RTC/1.0  
 Content-Type: application/sdp  
 Content-Length: xxx

【0083】本発明の原理を適用することができる多くの可能な実施形態があることを考慮すれば、図面に関して本明細書に記載した実施形態は例示的なものであって、本発明の範囲を限定するものと解釈してはならないことが認識される。例えば、ソフトウェア（またはハードウェア）として示した記載の実施形態の要素をハードウェア（またはソフトウェア）として実装することができること、または本発明の趣旨から逸脱することなく記載の実施形態の配置および詳細を修正することができることを当業者は認識しよう。したがって本明細書に記載した発明は、前記請求項およびその等価物の範囲に含まれる全ての実施形態を企図する。

【0084】

【発明の効果】ケルベロスプロトコル、NTLMプロトコルなどのセキュリティ機構を、SIPシグナリングオペレーションのメッセージフローに組み込んで、SIPクライアントとSIPプロキシが相互に認証できるようにするための方式が提供される。

【図面の簡単な説明】

【図1】本発明を実現することができる例示的なコンピュータシステムを概略的に示すブロック図である。

【図2】セッションシグナリング段階中に相互認証するSIPクライアントとSIPプロキシサーバを含むセッションイニシエーションプロトコル（SIP）システムを示す概略図である。

50 【図3】SIPクライアントとSIPプロキシサーバの

間の認証目的のシグナリングメッセージ交換を示す概略図である。

【図4】SIPのフレームワークに組み込まれたセキュリティ機構のオペレーションを表す状態機械を示す概略図である。

【図5】SIPクライアントが複数のSIPプロキシと認証オペレーションを実行するためのシグナリングメッセージ交換を示す概略図である。

【図6】ケルベロスセキュリティ機構を使用したSIPクライアントとプロキシの間のプレ認証プロセスにおけるメッセージフローを示す概略図である。

【図7】ケルベロスセキュリティ機構を使用したSIPクライアントとプロキシの間のチャレンジド認証プロセスにおけるメッセージフローを示す概略図である。

【図8】NTLMセキュリティ機構を使用したSIPクライアントとプロキシの間のチャレンジド認証プロセスにおけるメッセージフローを示す概略図である。

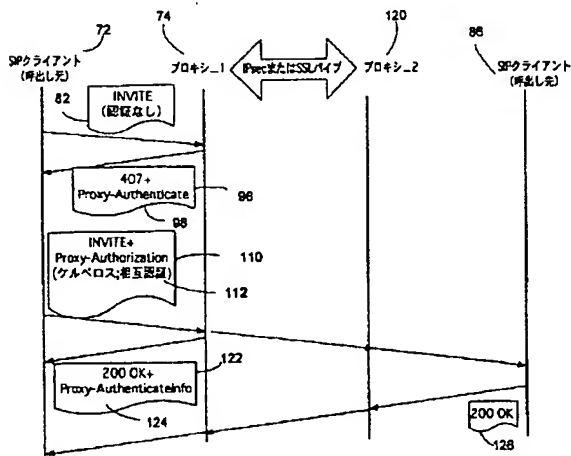
【図9】NTLMセキュリティ機構を使用したSIPクライアントとプロキシの間のプレ認証プロセスにおけるメッセージフローを示す概略図である。

#### 【符号の説明】

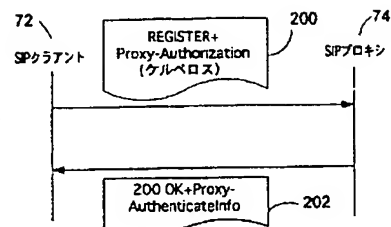
- 20 パーソナルコンピュータ
- 21 処理ユニット
- 22 システムメモリ
- 23 システムバス
- 24 ROM
- 25 RAM
- 27 ハードディスクドライブ
- 28 磁気ディスクドライブ
- 30 光ドライブ
- 32 ハードディスクドライブインタフェース
- 33 磁気ディスクドライブインタフェース

- \* 34 光ディスクドライブインタフェース
- 35 オペレーティングシステム
- 36 アプリケーションプログラム
- 37 その他のプログラムモジュール
- 38 プログラムデータ
- 40 キーボード
- 42 マウス
- 46 シリアルポートインタフェース
- 47 モニタ
- 48 ビデオアダプタ
- 49 リモートコンピュータ
- 50 メモリ記憶装置
- 51 ローカルエリアネットワーク
- 52 ワイドエリアネットワーク
- 53 ネットワークインタフェース
- 54 モデム
- 72 SIPクライアント
- 74 SIPプロキシサーバ
- 76、80 ユーザ
- 82 INVITEメッセージ
- 86 SIPクライアント
- 88 ワークステーション
- 90 シグナリング経路
- 96 407チャレンジメッセージ
- 102 ドメインコントローラ
- 108 サーバチケット
- 110 INVITEメッセージ
- 116 セッションキー
- 120 SIPプロキシサーバ
- 122 200OKメッセージ
- 170、172、176 KDC
- \* 180、182 ケルベロスモジュール

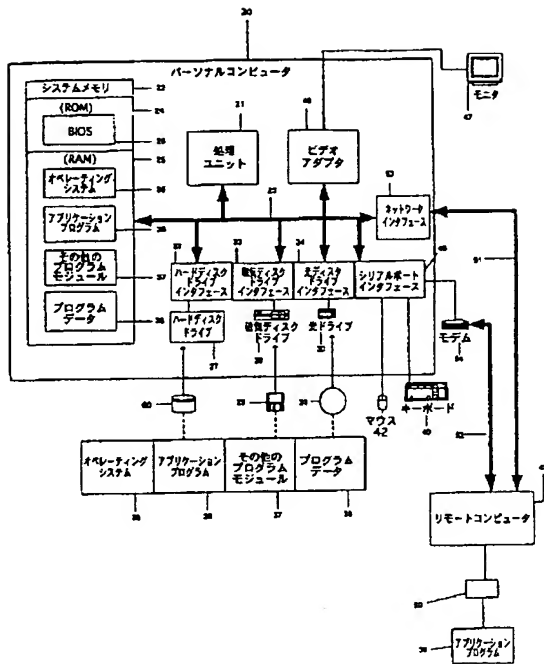
【図3】



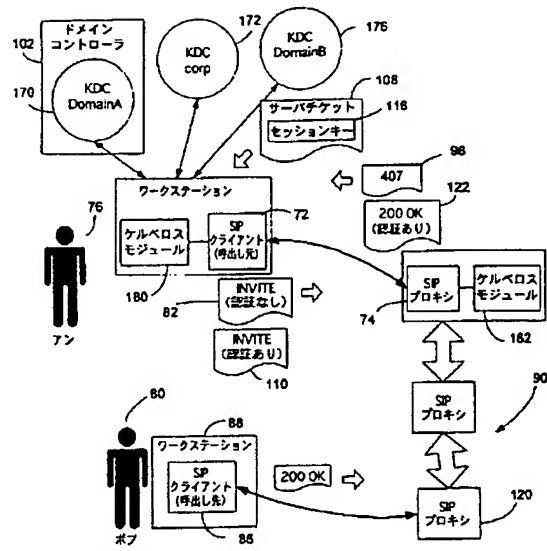
【図6】



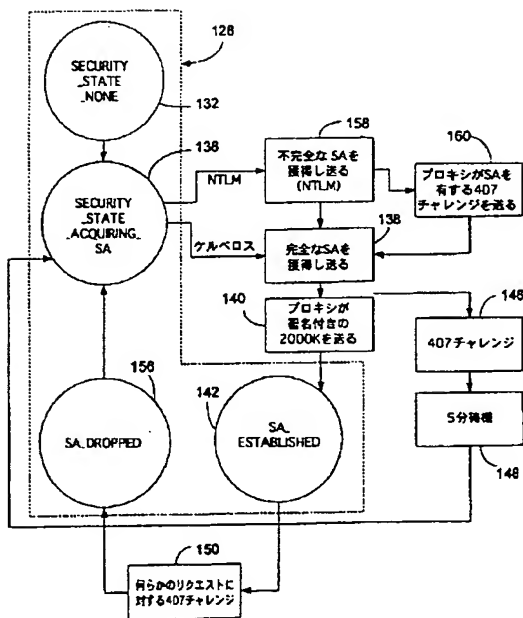
【図1】



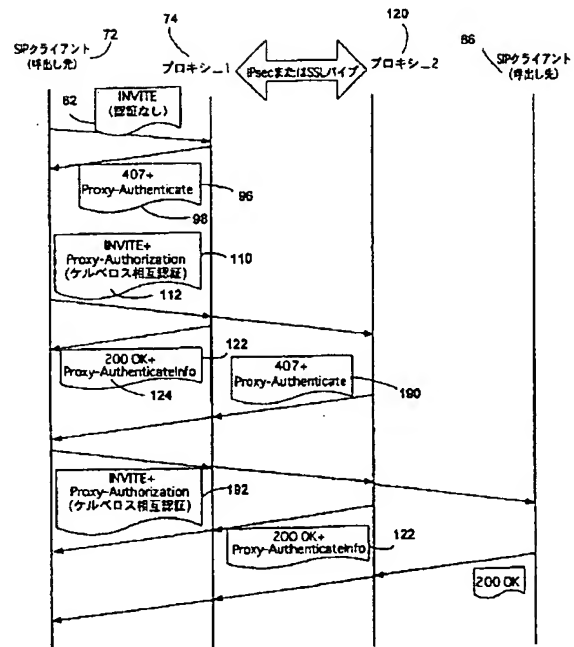
【図2】



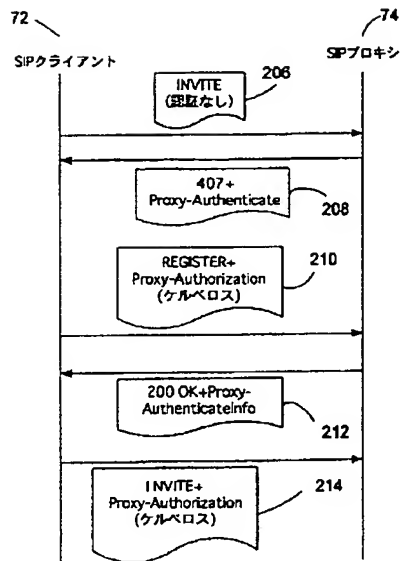
【図4】



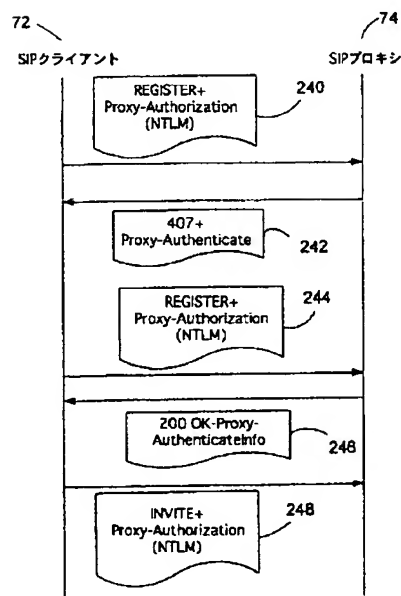
【図5】



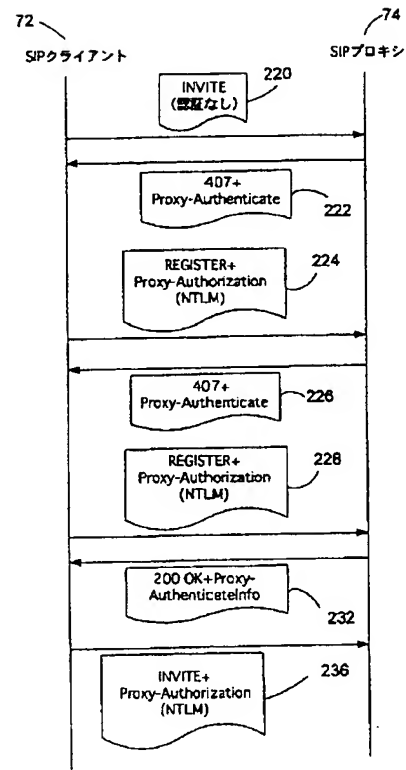
【図7】



【図9】



【図8】



フロントページの続き

(72)発明者 アン デミアジス  
 アメリカ合衆国 98052 ワシントン州  
 レッドモンド ノースイースト 67 スト  
 リート 14811

(72)発明者 ムー ハン  
 アメリカ合衆国 98052 ワシントン州  
 レッドモンド 153 アベニュー ノース  
 イースト 7204

Fターム(参考) 5B085 AE01 BA06 BG02 BG07  
 5J104 AA07 KA01 PA07